

Integration of convolutional neural network and extreme gradient boosting for breast cancer detection

Endang Sugiharti, Riza Arifudin, Dian Tri Wiyanti, Arief Broto Susilo

Department of Computer Science, Faculty of Mathematics and Natural Science, Universitas Negeri Semarang, Semarang, Indonesia

Article Info

Article history:

Received Apr 12, 2021

Revised Dec 4, 2021

Accepted Mar 13, 2022

Keywords:

Breast cancer
Convolutional neural network-
eXtreme gradient boosting
Data augmentation
Histopathology images
Transfer learning

ABSTRACT

With the most recent advances in technology, computer programming has reached the capabilities of human brain to decide things for almost all healthcare systems. The implementation of convolutional neural network (CNN) and extreme gradient boosting (XGBoost) is expected to improve the accurateness of breast cancer detection. The aims of this research were to; i) determine the stages of CNN-XGBoost integration in diagnosis of breast cancer and ii) calculate the accuracy of the CNN-XGBoost integration in breast cancer detection. By combining transfer learning and data augmentation, CNN with XGBoost as a classifier was used. After acquiring accuracy results through transfer learning, this research connects the final layer to the XGBoost classifier. Furthermore, the interface design for the evaluation process was established using the Python programming language and the Django platform. The results: i) the stages of CNN-XGBoost integration on histopathology images for breast cancer detection were discovered. ii) achieved a higher level of accuracy as a result of the CNN-XGBoost integration for breast cancer detection. In conclusion, breast cancer detection was revealed through the integration of CNN-XGBoost through histopathological images. The combination of CNN and XGBoost can enhance the accuracy of breast cancer detection.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Endang Sugiharti

Department of Computer Science, Faculty of Mathematics and Natural Science, Universitas Negeri Semarang

Jalan Raya Sekaran Gunung Pati Semarang, Indonesia

Email: endangsugiharti@mail.unnes.ac.id

1. INTRODUCTION

In almost every field, including healthcare, computer programming has now surpassed the human brain's ability to make decisions. This means that the advancement of programming in the field of computers will be used by humans to uphold human dignity itself. Now, the human brain's ability to be imitated by computers, particularly when it comes to solving problems that can be applied in the health care field. Through the latest technological advances, computer programming can be used to detect breast cancer. Data from the American Cancer Society (ACS) state that breast cancer is the main type of cancer in women. It was estimated that breast cancer was discovered in a total of 281,550 women and 43,600 died from breast cancer in 2021 [1]. The contribution of study is expected to open public awareness to be careful about the dangers of cancer. Cancer diagnosis is a hot topic in the healthcare field. The descriptive of cancer characteristic details and cancer study information can be obtained through the advancement of information technology, software, and hardware. Previously, cancer detection tended to use the digital image method, case-based reasoning, as well as the certainty factor method. Now, several scientists turned to data mining technology and prediction of breast cancer by machine learning models to deal with the significantly rising cancer feature data and

knowledge. Machine learning is a computer strategy for enhancing performance or generating accurate forecasts that is based on prior knowledge [2]. Using the feedback signal as instruction, machine learning seeks for an essential model of some raw data in a chosen probability space. It is feasible to automate a wide range of cognitive tasks using these concepts, from voice commands to operating an automobile [3]. Automation of the process of finding features or effective representations for machine learning on various tasks, including automatically transferring knowledge from one task to another simultaneously can be done using deep learning [4]. According to LeCun *et al.* [5], a system learns to classify immediately from pictures, texts, or voice. After a computer has been trained to recognize patterns in a great amount of labeled data, it converts an image's intensity values into an input space or feature vector, which classifiers can be used to find and categorize patterns in the input.

Computer models can undertake classification algorithms from text, pictures, or audio using one of the machine learning methodologies called deep learning. The models are built using a convolutional neural network (CNN) architecture with numerous layers and a huge number of datasets. Deep learning is employed in medical imaging to automatically detect cancer cells [6]. CNN is a sort of deep learning that is often employed in picture data, according to [7], [8]. CNN could be used to identify and recognize objects in images. In general, CNN isn't too distinct from previous neural networks. According to [9] CNN is composed primarily of significant layers known as convolutional layers. These convolutional layers are constructed from a fundamental structure block known as a convolution. A convolution is used to test the advantages of the pixels surrounding a small area of an image and modify it into nothing more than a single pixel.

A CNN can include dozens or even hundreds of layers, each learning to recognize distinct images. In image processing, every train picture is given a different resolution, and the output of each image is analysed and utilized as input to the next layer. Image processing can be begun with a step formula such as intensity and edges or complexity. According to the thickness of the layer, a function specifies an item uniquely [10], [11]. Because of the increased attention in deep learning, CNN has become the most frequently used approach for image analysis and classification. CNN has generated cutting-edge results in a variety of classification methods [12]. However, the deficiency of the dataset used may have an impact on the accuracy of CNN. The expected novelty in this issue can be resolve by adjusting the quantity of data used in the training process in order to enhance the level of accuracy. This is also in line with the opinion of [13]. To enhance the training samples [14], [15] used data augmentation. Data augmentation is a method of processing image data, augmentation is the process of altering or adjusting an image in such a way that the machine senses that the modified image is a different image, but people can still recognize that the modified image is the same image. This is a technique for increasing the number of images in a dataset by transforming the original images. Utilizing data augmentation improves the performance of the proposed models, particularly for classes that are still incorrectly classified [16].

Another option for overcoming the dataset's inadequacy, according to [17]-[19], is to use transfer learning. Transfer learning is a concept for utilizing previously learned characteristics of a big data set and then moving and implementing that learning to its dataset upon this [20] have succeeded in using the transfer learning and be able to increase accuracy by 18.3%. Aside from that, transfer learning can successfully resolve the issue of computing time and a small training dataset. Ren *et al.* [21] improved the performance of the modified national institute of standards and technology (MNIST) and Canadian institute for advanced research (CIFAR-10) datasets using CNN as the extraction of features and extreme gradient boosting (XGBoost) as a predictor, with MNIST's accuracy being 99.22% and CIFAR-10's being 80.77%.

XGBoost is a robust and effective machine learning technique for tree boosting that has been widely used in numerous disciplines to obtain state-of-the-art outcomes on multiple data issues [22]. The gradient booster algorithm is a quick implementation of XGBoost, getting the advantages of good speed and high precision. Scalability in different scenarios is the most significant factor behind XGBoost's success. This scalability is due to the preceding algorithm being optimized. A novel tree learning algorithm for handling sparse data is included in the innovations provided. This success was proven when XGBoost became one of the techniques widely used in machine learning in various cases [23]. At the top level, this XGBoost classifier has been applied to CNN and will generate image classification results. Based on the above background, the formulation of the problem posed by the study in this article is being as. How is the application of transfer learning and data augmentation with XGBoost classifier in CNN in the process of early detection of breast cancer?. What is the level of accuracy of the results obtained for the process of early detection of breast cancer with CNN by utilizing transfer learning and XGBoost classifier?. In this article, the novelty introduced is the use of machine learning models using CNN and XGBoost as classifier, where the application was developed for testing by using Django framework, Python language, hypertext markup language (HTML), cascading style sheets (CSS) Bootstrap, library of Tensorflow 2.0, and Keras 2.3.1. The expected contribution of this strategy is that using the data augmentation, transfer learning, and the XGBoost classifier in CNN can improve

the accuracy of breast cancer detection in the residual networks (ResNet50) architectural model, which can be used as a reference by other researchers who conduct breast cancer detection studies.

2. THEORETICAL BASIS

According [24], the CNN-XGBoost learning algorithm is described as follows. Let $X = \{(x_j, y_j) | 1 \leq j \leq M\}$, M denotes the training data set size, $x_j = [x_1, x_2, \dots, x_N]$ to be a collection of N feature vectors in \mathbb{R}^N or $\mathbb{R}^{\sqrt{N} \times \sqrt{N}}$ and vector x_j 's label is y_j .

- Begin by populating the training dataset, $X = \{(x_j, y_j) | 1 \leq j \leq M\}$.
- Increase the number of components in each training dataset x_j , if required, as a result of which the new data item can be shaped into a square matrix of mentions, $\sqrt{N} \times \sqrt{N}$.
- Convert x_j tensor format, $(\sqrt{N}, \sqrt{N}, z^{(l)})$.
- Configure the convolutional parameters for learning features: the quantity of convolutional layers, L ; convolutional layer yield depth, z ; define the filter sizes for every layer., $K^{(l)}$, and strides for filtering, $S_k^{(l)}$.
- Determine the convolutions for every layer l , in $1 \dots L$ to produce the $y_i^{(l)}$ for layer, l :

$$y_i^{(l)} = \phi \left(B_i^{(l)} + \sum_{j=1}^{f^{(l-1)}} K_{ij}^{(l)} * y_j^{(l-1)} \right)$$

- Reshape $y_i^{(l)}$ to a vector of lengths $(\sqrt{N}, \sqrt{N}, z^{(l)})$ – $y y^{(l)}$
- Create a fresh training data for the class prediction layer

$$X_{new} = \{(y y_j, y_j) | 1 \leq j \leq M\}$$

- Set the parameters for the prediction phase: the overall number of trees, K_K ; parameters of normalization, as well as γ and λ ; parameter of column subsampling; the maximum tree depth and; rate of learning.
- Evaluate the output class labels:

$$\hat{y}_i = \phi(Y Y_i) = \sum_{k=1}^{K_K} f_k(x_i), \quad f_k \in F,$$

$$\text{Where } F = f(Y Y_i) = w_{q(Y Y_i)}(q : \mathbb{R}^N \rightarrow T, w \in \mathbb{R}^T).$$

- Determine the optimal leaf weight for the finest tree structure.

$$w_j^* = - \frac{\sum_{i \in l_j} g_i}{\sum_{i \in l_j} h_i + \lambda}$$

- Using the scoring function, compute the tree structure's quality, q .

$$\tilde{\mathcal{L}}^{(t)}(q) = - \frac{1}{2} \frac{(\sum_{i \in l_j} g_i)^2}{\sum_{i \in l_j} h_i + \lambda} + \gamma T$$

T denotes the amount of leaves on the tree.

- Determine the finest splitting points

$$\mathcal{L}_{split} = \frac{1}{2} \left[\frac{(\sum_{i \in l_L} g_i)^2}{\sum_{i \in l_L} h_i + \lambda} + \frac{(\sum_{i \in l_R} g_i)^2}{\sum_{i \in l_R} h_i + \lambda} + \frac{(\sum_{i \in l} g_i)^2}{\sum_{i \in l} h_i + \lambda} \right] - \gamma$$

- Terminate.

The algorithm of CNN-XGBoost has time complexity: $\mathcal{O}Ld^2mnpq + \mathcal{O}r(Kt + \log B)$ which reduces to $\mathcal{O}Ld^2mnpq$, where L represents the number of layers, the number of input or output channels is denoted

by d , the size of the data matrix is $m \times n$, the size of filter is $p \times q$, $r = \|x\|$ is the amount of entries that are not missing, the amount of trees is K , the tree depth is t , and the length of the block is B .

3. METHODOLOGY

The research methodology used as a reference in conducting research is as follows.

a. Literature study

At this stage, materials, information and theory are collected in books as well as references from articles, journals, and other scientific works related to the object of research and the algorithms used in research.

b. Data collection

Data collection was carried out to obtain the data used in the study. The dataset used in the study was histopathology breast cancer consisting of 162 images of breast cancer specimens (BCa) that were scanned 40 times. From the extraction results 277,524 patches measuring 50 x 50 (198,738 invasive ductal breast cancer (IDC) negative and 78,786 positive IDC). So that in this study used 2 classes, namely negative and positive.

c. Research stages

– Data augmentation stages

Data augmentation is done in real-time during training on CNN. After data augmentation can provide many variations to CNN and increase the amount of relevant data [25]. This research uses ImageDataGenerator from Keras to augment data. ImageDataGenerator is performed during the training process. By using this type of data augmentation the researcher wants to ensure that the network, when trained, will get new variations in each epoch. The process of applying on-the-fly augmentation with the following flow: a) image data generator receives a batch (collection) of images to be used for training, b) image data generator takes that image set and then applies a series of random changes to each image in the batch (including random rotation, resizing, and shifting), c) replacing the original batch with a new batch that was changed randomly, d) train CNN on randomly changed batches (the original data itself is not used for training). In each epoch, ImageDataGenerator applies transformations to existing images and transforms them for training. The number of pictures in each epoch is the same as the number of original images. This research will use various kinds of transformations such as `featurewise_center`, `rotation_range` `width_shift_range`, `height_shift_range`, `horizontal_flip`. Transformations will be added as accuracy increases.

– Transfer learning stages

The usage of knowledge from heretofore trained models to accomplish new tasks is regarded to as transfer learning. As a feature extractor, transfer learning will be used in this research. The transfer learning process can reduce training time, computations, and the use of available hardware resources [20].

– XGBoost stages

In this study, XGBoost was used for the classification stage. After the CNN training stage, XGBoost replaces the output layer, replaces CNN's softmax classifier, and uses CNN's trainable features for training. Lastly, by testing images, the CNN-XGBoost model obtains new classification findings [21].

– System planning

In this study, the system design was carried out as a test tool for the application of the CNN technique using transfer learning and data augmentation with the XGBoost classifier in the early detection of breast cancer.

4. RESULTS AND DISCUSSION

4.1. Results obtained

In this part, the researcher divided the stages, namely the model concept and system implementation. The model concept describes the stage of making the model get accurate results and finally saving the model that will be used in system implementation. The use of data augmentation and transfer learning in the CNN method to enhance the accurateness of breast cancer detection has several stages of research. At this point, the authors develop a flow concept that will be applied to the CNN method via a combination of data augmentation and transfer learning. In this research, the type of transfer learning applied to the CNN as an image feature extraction process in the image dataset is ResNet50.

The process of extracting the breast cancer histopathology feature is carried out using a Google colaboratory server with GPU runtime settings connected to Google Drive as storage media. The author performs the image feature extraction process by extracting all images in the image dataset into a fixed-size vector. The image dataset feature extraction process is carried out using the ImageNet architecture, namely ResNet50 to create the ResNet50 model. The model will also use ImageDataGenerator as a function for the

data augmentation method during training. After gaining accuracy results through transfer learning, this research links the final layer to the XGBoost classifier.

The first step, to run the modeling process on Google Colab is installing Kaggle on Google Colab, followed by uploading the Kaggle token API so that the dataset from Kaggle can be downloaded. The dataset is divided into 2 sets, namely training and testing with a ratio of 70:30. This data division uses the Sklearn library using the train test split function with the number of tests splits worth 0.3. The way the train test split works is to take test data on the entire dataset up to 30% and use the rest as training data. The results of data division are training data of 194,266 and testing data of 83,258 with details as shown in Table 1.

Table 1. Details of test data and training data

| | Positive | Negative |
|---------------|----------|----------|
| Training data | 55.150 | 139.116 |
| Testing data | 23.636 | 59.622 |

Balancing the training dataset is carried out so that the model training process can learn optimally later. In this research, data augmentation was carried out on positive class training data from 55,150 to 139,125 using ImageDataGenerator. Making train and test batches is used to facilitate the training stage. This study using the number of train batches of 500 for training and validation. The system executes data augmentation which in this research uses Image Data Generator owned by the Keras library. Three arguments are used to augment the data., horizontal flip (true), vertical flip (true), and fill mode (nearest). The number of images that will be generated by image data generator follows the number of epochs. Figure 1 shows sample images before and after the data augmentation process.

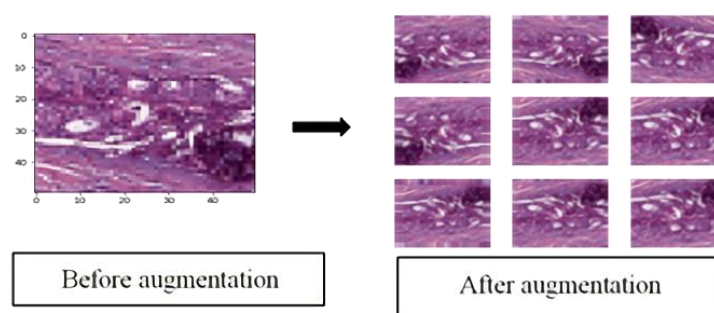


Figure 1. Image results before and after augmentation

The next step is transfer learning. This work employs the ResNet50 architectural model, which was trained on ImageNet data by constructing an architecture that is similar to ResNet50 but lacks the full linked layer and downloads weight. Keras provides several popular ImageNet models. ResNet50 weights can acknowledge colors and textures because they were trained on ImageNet datasets. As a result, the ResNet50 weights are applied to all images in the dataset to extract features. The input model value was calculated from the feature extraction process's final vector, which is 2048. The next process is training and data validation. In the training process, the data used is 70% of the data taken randomly, the rest of the train test split with a test size of 0.3. At the training stage, Adam's optimizer was used to complete the task with 10 epoch training models, as seen in Figure 2.

The model was trained using the ResNet50 architecture. The loss and accuracy values from the model were measured on the training and validation datasets by testing the 10 epoch model. The model trained by CNN is assessed using just a graph as shown in Figure 3 after it has been effectively trained and has received the outcomes of training and validation accuracy.

```

Epoch 1/10
389/389 [=====] - ETA: 0s - loss: 0.2832 - accuracy: 0.8813
Epoch 00001: val_loss improved from inf to 0.32441, saving model to breast_histopathology_transfer_best.hdf5
389/389 [=====] - 325s 837ms/step - loss: 0.2832 - accuracy: 0.8813 - val_loss: 0.3244 - val_accuracy: 0.8600
Epoch 2/10
389/389 [=====] - ETA: 0s - loss: 0.2400 - accuracy: 0.8979
Epoch 00002: val_loss improved from 0.32441 to 0.31377, saving model to breast_histopathology_transfer_best.hdf5
389/389 [=====] - 245s 631ms/step - loss: 0.2400 - accuracy: 0.8979 - val_loss: 0.3138 - val_accuracy: 0.8659
Epoch 3/10
389/389 [=====] - ETA: 0s - loss: 0.2320 - accuracy: 0.9015
Epoch 00003: val_loss improved from 0.31377 to 0.30978, saving model to breast_histopathology_transfer_best.hdf5
389/389 [=====] - 239s 615ms/step - loss: 0.2320 - accuracy: 0.9015 - val_loss: 0.3098 - val_accuracy: 0.8671
Epoch 4/10
389/389 [=====] - ETA: 0s - loss: 0.2254 - accuracy: 0.9044
Epoch 00004: val_loss improved from 0.30978 to 0.30424, saving model to breast_histopathology_transfer_best.hdf5
389/389 [=====] - 239s 614ms/step - loss: 0.2254 - accuracy: 0.9044 - val_loss: 0.3042 - val_accuracy: 0.8702
Epoch 5/10
389/389 [=====] - ETA: 0s - loss: 0.2232 - accuracy: 0.9059
Epoch 00005: val_loss improved from 0.30424 to 0.30019, saving model to breast_histopathology_transfer_best.hdf5
389/389 [=====] - 239s 614ms/step - loss: 0.2232 - accuracy: 0.9059 - val_loss: 0.3002 - val_accuracy: 0.8727
Epoch 6/10
389/389 [=====] - ETA: 0s - loss: 0.2185 - accuracy: 0.9072
Epoch 00006: val_loss did not improve from 0.30019
389/389 [=====] - 238s 613ms/step - loss: 0.2185 - accuracy: 0.9072 - val_loss: 0.3003 - val_accuracy: 0.8716
Epoch 7/10
389/389 [=====] - ETA: 0s - loss: 0.2190 - accuracy: 0.9070
Epoch 00007: val_loss improved from 0.30019 to 0.29782, saving model to breast_histopathology_transfer_best.hdf5
389/389 [=====] - 239s 613ms/step - loss: 0.2190 - accuracy: 0.9070 - val_loss: 0.2978 - val_accuracy: 0.8729
Epoch 8/10
389/389 [=====] - ETA: 0s - loss: 0.2177 - accuracy: 0.9070
Epoch 00008: val_loss did not improve from 0.29782
389/389 [=====] - 238s 613ms/step - loss: 0.2177 - accuracy: 0.9070 - val_loss: 0.3021 - val_accuracy: 0.8713
Epoch 9/10
389/389 [=====] - ETA: 0s - loss: 0.2159 - accuracy: 0.9091
Epoch 00009: val_loss improved from 0.29782 to 0.29432, saving model to breast_histopathology_transfer_best.hdf5
389/389 [=====] - 239s 614ms/step - loss: 0.2159 - accuracy: 0.9091 - val_loss: 0.2943 - val_accuracy: 0.8753
Epoch 10/10
389/389 [=====] - ETA: 0s - loss: 0.2135 - accuracy: 0.9100
Epoch 00010: val_loss improved from 0.29432 to 0.29321, saving model to breast_histopathology_transfer_best.hdf5
389/389 [=====] - 239s 614ms/step - loss: 0.2135 - accuracy: 0.9100 - val_loss: 0.2932 - val_accuracy: 0.8748

```

Figure 2. Accuracy result of training and validation of ResNet50 model



Figure 3. Graph of training and validation accuracy and training and validation loss model of ResNet 50

Figure 4 is the result of the classification accuracy generated by the training and validation of the ResNet50 model, with an accuracy of 88%. After the model goes through the training and validation process, the next part is to take the last part of the model layer to be integrated into the XGBoost classifier. After taking the last layer of the process model, the next step is to apply it to the XGBoost classifier using the library from XGBoost. With the XGBoost classifier, the model can produce a train and test score just after the fitting procedure. The results of accuracy using the XGBoost classifier are 92% for the training score and 90% for the test score. Furthermore, the model is analyzed using a confusion matrix, as in Figure 5.

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| negative | 0.91 | 0.92 | 0.91 | 59622 |
| positive | 0.79 | 0.76 | 0.77 | 23636 |
| accuracy | | | 0.88 | 83258 |
| macro avg | 0.85 | 0.84 | 0.84 | 83258 |
| weighted avg | 0.87 | 0.88 | 0.87 | 83258 |

Figure 4. Classification accuracy for the ResNet 50 model

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| Negative | 0.91 | 0.93 | 0.92 | 4961 |
| Positive | 0.93 | 0.90 | 0.92 | 5039 |
| accuracy | | | 0.92 | 10000 |
| macro avg | 0.92 | 0.92 | 0.92 | 10000 |
| weighted avg | 0.92 | 0.92 | 0.92 | 10000 |

Figure 5. Training features report for ResNet 50 model with XGBoost classifier

The conclusion of this study can be seen in the results table of loss and accuracy values as in Table 2. Table 2 shows that XGBoost affects increasing the accuracy of the ResNet50 model. The results of the accuracy of CNN validation with ResNet50 have increased after using the XGBoost classifier from 88% to 90%, while accuracy. The high value of this accuracy is influenced by the image quality which is very adequate and good.

Table 2. Results of loss and accuracy values

| | Training data accuracy | Validation data accuracy |
|------------------|------------------------|--------------------------|
| CNN+ResNet50 | 0.91 | 0.88 |
| CNN+ResNet50+XGB | 0.92 | 0.90 |

4.2. System implementation stage

At this point, the software is being prepared to test the approaches used in this research. The user interfaces design for this program was created using HTML, CSS, and Javascript programming languages. Meanwhile, the Django platform and the Python language are used for analysis. The results from the previous training model will be saved into a file with extension of .h5 which will be used later in the application. This application has 3 menus, namely home, what is, and about us. On the home menu, there is a description of the research title and the method used. The display design of the home menu can be seen in Figure 6.



Figure 6. Home application display

After selecting "predict now" will move the screen to the image upload display. In the upload image view, select the image to be classified. The image upload display can be seen in Figure 7. After selecting the image you want to classify, then select "predict", the prediction results will appear as in Figure 8.

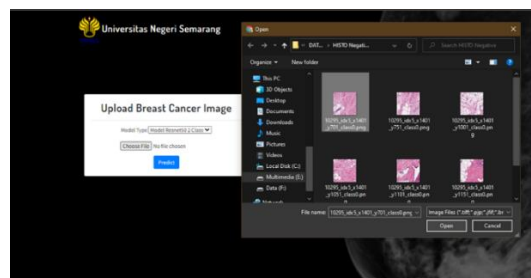


Figure 7. Display for image uploading

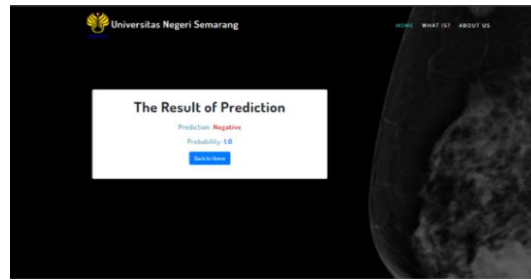


Figure 8. Display of prediction results

4.3. Discussion

This study applies transfer learning and the XGBoost classifier to optimize the CNN to improve the accuracy of breast cancer detection using histopathology images. Based on the results of the implementation of a combination of transfer learning and the XGBoost classifier on the CNN that has been carried out, it can be known the accuracy improvement of CNN-XGBoost in diagnosing breast cancer. The dataset is obtained from Kaggle with a class distribution graph as shown in Figure 9.

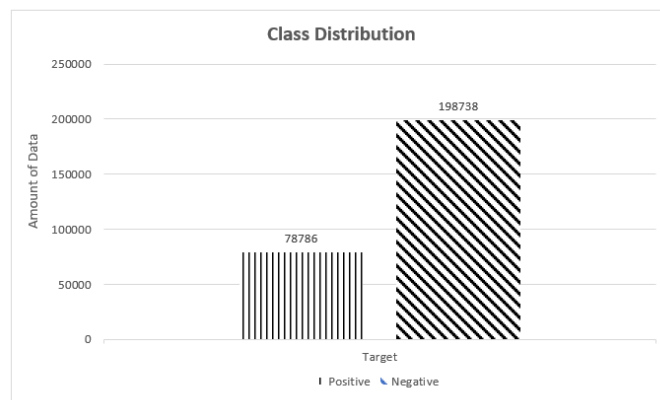


Figure 9. Class distribution dataset

Figure 9 also shows that the dataset is very unbalanced in classes 0 and 1, where 1 for the class is positive and 0 for a class is negative. The vast difference in the amount of data can affect the training process, where the model will learn more about negative classes than positive classes. For this reason, this study conducted balancing training data. The data was added to the positive class. A total of 55,150 new data were generated using ImageDataGenerator, so this study used 2 data augmentation processes. The first is data augmentation for the balancing training data process and the second for the training process. The training data balancing stage is carried out after the distribution of training data and test data with details of 30% for testing data and 70% for training data.

The next stage is to create train and test batches for the training process. After making the number of batches used, the next step is training and validation using the ResNet50 model, this model has been previously trained or commonly known as transfer learning. Transfer learning used in this study functions as feature extractors and the weights used is derived from ImageNet [24]. After building the transfer learning model, the model is compiled so that it can be used for the training process later.

In this process, we arrange the model to be ready for the training process. Where the variables used are as follows: 1) loss, the method of measuring the loss value based on categorical values; 2) optimizer is an algorithm for updating weights and biases in the learning process of artificial neural networks, to minimize errors or the difference between the network output and target. The Adam optimizer is a combination of the RMSProp and momentum optimizer, which does have several advantages, including being computationally efficient, memory efficient, and suitable for various optimization problems in the field of machine learning; 3) metrics, the measured value of the matrix in this study uses the accuracy value as the measurement value.

The next stage, data augmentation is used to provide a new picture of the dataset during the training process later. By using data augmentation, the dataset to be trained later will not be the same as the dataset at first, data augmentation will train the model with new data according to the functions used in each batch. This data augmentation is done using ImageDataGenerator from the Keras package [3] comparing and analyzing data augmentation methods in image classification, and the results can be concluded that traditional transformation has more impact than other augmentation methods. The data augmentation function used in this study is as shown in Table 3.

Table 3. Arguments and data augmentation functions

| Argument | Function | Amount |
|-----------------|-----------------------------|---------|
| Horizontal Flip | Flip the image horizontally | True |
| Vertical Flip | Flip the image vertically | True |
| Fill mode | Fill in the empty area | Nearest |

Based on Table 3 in data augmentation, will go through 3 arguments for each image, each image will be slightly different from the others due to horizontal and vertical flip. This will help the built model to recognize a large number of images, thus making them more efficient. This data augmentation works during the training process, the image will be augmented in real-time during the training process. Data augmentation is not carried out in the validation process because the validation process is a process to validate a model that has been trained with the original image so it is not good for changing the image in the validation process. An example of an image result that has gone through data augmentation can be seen in Figure 10.

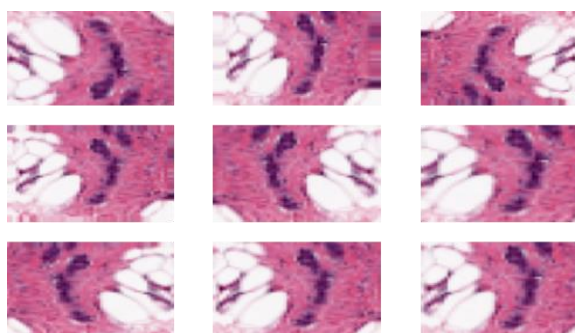


Figure 10. Image results after going through data augmentation

The next stage is training and validation to get accurate results in the transfer learning model. Then the model that has been trained is taken to the last layer and then applied to the XGBoost Classifier. XGBoost classifier is tasked to classify models that have been previously trained by ResNet50 so that they get new training and validation accuracy results.

The validation accuracy has increased by 2% when utilizing the XGBoost classifier, from 88% to 90%, as demonstrated in the training and validation results. Furthermore, the model is stored for use in the test system. This is further confirmed by the findings of [7], which found that applying transfer learning using GoogleNet architectural can increase the model accuracy. The contribution of this approach is that it has been established that using data augmentation, transfer learning and the XGBoost classifier in the CNN method can improve the accurateness of breast cancer detection in the ResNet50 architectural model, and it can be used as a reference by other researchers conducting breast cancer detection studies.

5. CONCLUSION

The following conclusions are drawn from the obtained results and their explanation as given as; i) the following phases of CNN-XGBoost implementation were discovered via data augmentation and transfer learning. Starting with the collection of the dataset, followed by the preprocessing stage, dividing the data into training data and testing data with details of 70% training data and 30% for testing data then data augmentation is carried out using ImageDataGenerator from the Keras package, transfer learning used in this study functions as feature extractors and weights used are from ImageNet. The next stage is the training and validation stage to get the accuracy results for each model. Then the model that has been trained is taken to

the last layer and then applied to the XGBoost classifier. XGBoost classifier is tasked to classify models that have been previously trained by ResNet50 so that they get new training and validation accuracy results. Then the model is stored for use in the test system; ii) the accuracy of validation results of the integration of CNN-XGBoost on histopathological images for breast cancer detection has increased by 2% from 88% to 90%. This research gives the directions for future research, which can be given are being as: i) the detection accuracy of this breast cancer will increase along with the image quality used so that it can be tried on other types of images and ii) to get better results, it can also be done by increasing the number of datasets, increasing data augmentation by adding any different parameter or trying other architectural models.

ACKNOWLEDGEMENTS




Our gratitude goes to the Research and Community Service Institution of Universitas Negeri Semarang for funding this research.

REFERENCES




- [1] R. L. Siegel, K. D. Miller, H. E. Fuchs, and A. Jemal, "Cancer Statistics, 2021," *CA. Cancer J. Clin.*, vol. 71, no. 1, pp. 7–33, January 2021, doi: 10.3322/caac.21654.
- [2] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning second edition*, MIT press, no. 1. 2018.
- [3] F. Chollet, *Deep Learning mit Python und Keras: Das Praxis-Handbuch vom Entwickler der Keras-Bibliothek*, MITP-Verlags GmbH & Co. KG, 2018.
- [4] L. Deng and D. Yu, "Deep learning: Methods and applications," *Found. Trends Signal Process.*, vol. 7, no. 3–4, pp. 197–387, 2014, doi: 10.1561/20000000039.
- [5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.
- [6] S. Charan, M. J. Khan, and K. Khurshid, "Breast cancer detection in mammograms using convolutional neural network," *2018 Inter. Conf. on Computing, Mathematics and Engineering Technologies.*, 2018, pp. 1–5, doi: 10.1109/ICOMET.2018.8346384.
- [7] E. C. Djamal, R. I. Ramadhan, M. I. Mandasari, and D. Djajasmita, "Identification of post-stroke eeg signal using wavelet and convolutional neural networks," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 5, pp. 1890–1898, October 2020, doi: 10.11591/eei.v9i5.2005.
- [8] M. A. Rasyidi and T. Bariyah, "Batik pattern recognition using convolutional neural network," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 4, pp. 1430–1437, August 2020, doi: 10.11591/eei.v9i4.2385.
- [9] D. N. V. S. L. Indira, J. Goddu, B. Indira, V. M. L. Challa, and B. Manasa, "Materials Today : Proceedings A review on fruit recognition and feature evaluation using CNN," *Mater. Today Proc.*, pp. 1–6, July 2021, doi: 10.1016/j.matpr.2021.07.267.
- [10] MathWorks, "Convolutional Neural Network," 2020. [Online]. Available: <https://www.mathworks.com/solutions/deep-learning/convolutional-neural-network.html>.
- [11] A. Fawzi, H. Samulowitz, D. Turaga, and P. Frossard, "Adaptive data augmentation for image classification," *2016 IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 3688–3692, doi: 10.1109/ICIP.2016.7533048.
- [12] S. N. M. Safuan, M. R. M. Tomari, W. N. W. Zakaria, M. N. H. Mohd, and N. S. Suriani, "Investigation of white blood cell biomarker model for acute lymphoblastic leukemia detection based on convolutional neural network," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 2, pp. 611–618, April 2020, doi: 10.11591/eei.v9i2.1857.
- [13] A. Mikołajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," *2018 International Interdisciplinary PhD Workshop (IIPHDW)*, 2018, pp. 117–122, doi: 10.1109/IIPHDW.2018.8388338.
- [14] L. Perez and J. Wang, "The Effectiveness of Data Augmentation in Image Classification using Deep Learning," *Convolutional Neural Networks Vis. Recognit*, vol. 11, pp. 1–8, 2017.
- [15] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, 2019, doi: 10.1186/s40537-019-0197-0.
- [16] O. A. Shawky, A. Hagag, E. S. A. El-Dahshan, and M. A. Ismail, "Remote sensing image scene classification using CNN-MLP with data augmentation," *Optik (Stuttg.)*, vol. 221, p. 165356, 2020, doi: 10.1016/j.ijleo.2020.165356.
- [17] L. H. S. Vogado, R. M. S. Veras, F. H. D. Araujo, R. R. V. Silva, and K. R. T. Aires, "Leukemia diagnosis in blood slides using transfer learning in CNNs and SVM for classification," *Eng. Appl. Artif. Intell.*, vol. 72, pp. 415–422, 2018, doi: 10.1016/j.engappai.2018.04.024.
- [18] M. H. Devare, "Challenges and Opportunities in High Performance Cloud Computing," *Res. Anthol. Archit. Fram. Integr. Strateg. Distrib. Cloud Comput.*, pp. 1989–2018, April 2019, doi: 10.4018/978-1-7998-5339-8.ch096.
- [19] S. Deepak and P. M. Ameer, "Brain tumor classification using deep CNN features via transfer learning," *Comput. Biol. Med.*, vol. 111, p. 103345, 2019, doi: 10.1016/j.compbiomed.2019.103345.
- [20] H. M. Ahmad, S. Ghuffar, and K. Khurshid, "Classification of Breast Cancer Histology Images Using Transfer Learning," *2019 16th International Bhurban Conf. on Appl. Sciences and Technology.*, 2019, pp. 328–332, doi: 10.1109/IBCAST.2019.8667221.
- [21] X. Ren, H. Guo, S. Li, and S. Wang, "A Novel Image Classification Method with CNN-XGBoost Model," pp. 378–390, August 2017, doi: 10.1007/978-3-319-64185-0.
- [22] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2016, pp. 785–794, doi: 10.1145/2939672.2939785.
- [23] N. A. Simanjuntak, J. Hendarto, and Wahyono, "The effect of image preprocessing techniques on convolutional neural network-based human action recognition," *J. Theor. Appl. Inf. Technol.*, vol. 98, no. 16, pp. 3364–3374, August 2020.
- [24] S. Thongsuwan, S. Jaiyen, A. Padcharoen, and P. Agarwal, "ConvXGB: A new deep learning model for classification problems based on CNN and XGBoost," *Nucl. Eng. Technol.*, vol. 53, no. 2, pp. 522–531, 2021, doi: 10.1016/j.net.2020.04.008.
- [25] N. E. M. Khalifa, M. Loey, and M. H. N. Taha, "Insect pests recognition based on deep transfer learning models," *J. Theor. Appl. Inf. Technol.*, vol. 98, no. 1, pp. 60–68, 2020, doi: 10.6084/m9.figshare.13271123.v3.

BIOGRAPHIES OF AUTHORS






Endang Sugiharti    received a master's degree in Computer Science from the University of Indonesia, in 2005. She is a lecturer in the Computer Science Department of Universitas Negeri Semarang. Her interests are in image processing, information system, and computer-aided instructions. She can be contacted at email: endangsugiharti@mail.unnes.ac.id.






Riza Arifudin    received a master's degree in Computer Science from the University of Gajah Mada, Indonesia, in 2010. He is a lecturer in the Computer Science Department of Universitas Negeri Semarang. His interests are in mobile application, information systems, and computer-aided instructions. He can be contacted at email: rizaarifudin@mail.unnes.ac.id.



Dian Tri Wiyanti    received a master's degree in Computer Science from the University of Gajah Mada, Indonesia, in 2012. She is a lecturer in the Mathematics Department of Universitas Negeri Semarang. Her interests are in information systems and machine learning. She can be contacted at email: dian.t.wiyanti@gmail.com.



Arief Broto Susilo    received a bachelor's degree in Computer Science from Universitas Negeri Semarang, Indonesia, in 2020. His interests are in information systems and image processing. He can be contacted at email: ariefbees@gmail.com.